Attorney Docket No.: 5577-277 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Carol Ann Jones et al.
Serial No.: 10/673,752
Filed: September 29, 2003
For: METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR CREATING USER INTERFACE TO APPLICATIONS USING GENERIC USER INTERFACE TEMPLATES

Confirmation No.: 1478
Group Art Unit: 2194
Examiner: Price, Nathan E.

September 4, 2007

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATION OF TRANSMISSION

I hereby certify that this correspondence is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4) to the U.S. Patent and Trademark Office on September 4, 2007.

Candi L. Riggs

Candi L. Riggs

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. § 41.37

Sir:

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" electronically transmitted on July 9, 2007

Real Party In Interest

The real party in interest is assignee International Business Machines, Inc., Armonk, New York.

Related Appeals and Interferences

Appellants are aware of no appeals or interferences that would be affected by the present appeal.

Status of Claims

Claims 1-6, 8-12, 14, 15, and 17-20 remain pending as of the filing date of this Brief, and stand finally rejected. Appellants appeal the final rejection of Claims 1-6, 8-12, 14, 15, and 17-20 in the final office action mailed April 10, 2007 (hereinafter "Final Action").

The attached Appendix A presents the claims at issue as finally rejected in the Final Action.

Status of Amendments

An Amendment filed January 8, 2007 has been entered. The attached Appendix A presents the pending claims and the corresponding status of each of the pending claims.

Summary of the Claimed Subject Matter

Some embodiments of the present invention according to independent Claim 1 provide methods of providing a user interface for an application. A user interface template that implements a task pattern for a model class is identified. *See, e.g.*, Fig. 2, user interface template 240 and description at page 8, lines 24-28. A model adapter that provides an interface for the application that conforms to the model class is identified. *See, e.g.*, Fig. 2, model adapter 232 and description at page 8, lines 29-31. A user interface for the application is created from the identified user interface template and the identified model adapter. *See, e.g.*, Fig. 2, user interface object 220 and description at page 8, lines 28-31.

In some embodiments of the present invention according to Claim 3, which depends from Claim 1, the user interface template defines an abstract portlet, and creating a user interface comprises creating a portlet instance that communicates with the application via the model adapter. *See, e.g.*, Fig. 3, abstract portlet 330 and portlet 320 and description at page 8, line 32 through page 9, line 2. In some embodiments of the present invention according to Claim 4, which also depends from Claim 1, creating a portlet instance comprises configuring the portlet instance using an application portlet builder portlet. *See, e.g.*, application portlet builder 310 and description at page 8, line 32 through page 9, line 2.

In some embodiments of the present invention according to independent Claim 8, a computer apparatus is configured to provide a user interface builder object operative to identify a user interface template that implements a task pattern for a model class. *See, e.g.*, Fig. 2, user interface builder object 210, user interface template 240 and description at page 8, lines 24-28. The computer apparatus is

further configured to identify a model adapter that provides an interface for an application that conforms to the model class. *See, e.g.*, Fig. 2, model adapter 232 and description at page 8, lines 29-31. The computer apparatus is further configured to create a user interface for the application from the identified user interface template and the identified model adapter. *See, e.g.*, Fig. 2, user interface object 220 and description at page 8, lines 28-31.

In some embodiments of the present invention according to Claim 10, which depends from Claim 8, the user interface template defines an abstract portlet, and the user interface builder is operative to create a portlet instance that communicates with the application via the model adapter. *See, e.g.*, application portlet builder 310 and description at page 8, line 32 through page 9, line 2. In some embodiments of the present invention according to Claim 11, which also depends from Claim 8, the user interface builder comprises an application portlet builder portlet operative to configure the portlet instance. *See, e.g.*, application portlet builder 310 and description at page 8, line 32 through page 9, line 2.

In some embodiments of the present invention according to independent Claim 14, a computer apparatus is configured to provide an application. *See, e.g.*, Fig. 3, application 350 and description at page 9, lines 2-4. The computer apparatus is also configured to provide a model adapter that provides an interface for the application that conforms to a model class. *See, e.g.*, Fig. 3, business object adapter 340 and description at page 9, lines 2-4. The computer apparatus is further configured to provide a portlet that implements a task pattern for the model class and that communicates with the application via the model adapter according to the interface. *See, e.g.*, Fig. 3, portlet 320 and description at page 8, line 32 through page 5, line 2.

In some embodiments of the present invention according to independent Claim 17, a computer program product for providing a user interface to an application includes computer readable code embodied in a computer readable storage medium, the computer readable code comprising program code configured to provide a user interface builder object operative to identify a user interface template that implements a task pattern for a model class (*See, e.g.*, Fig. 2, user interface template 240 and description at page 8, lines 24-28), to identify a model adapter that provides an interface for the application that conforms to the model class (see, *e.g.*, Fig. 2, model

adapter 232 and description at page 8, lines 29-31), and to create a user interface for the application from the identified user interface template and the identified model adapter (see, *e.g.*, Fig. 2, user interface object 220 and description at page 8, lines 28-31).

In some embodiments of the present invention according to Claim 19, which depends from Claim 17, the user interface template defines an abstract web portlet, and the code configured to provide a user interface builder comprises code configured to create a concrete web portlet instance that communicates with the application via the model adapter. *See, e.g.*, application portlet builder 310 and portlet 320 and description at page 8, line 32 through page 9, line 2. In some embodiments of the present invention according to Claim 20, which also depends from Claim 17, the code configured to create a concrete web portlet instance comprises code configured to provide an application portlet builder portlet operative to configure the portlet instance. *See, e.g.*, application portlet builder 310 and description at page 8, line 32 through page 9, line 2.

Grounds of Rejection to be Reviewed on Appeal

1. Are Claims 8-12, 14 and 15 properly rejected under 35 U.S.C. § 101 as being allegedly directed to non-statutory subject matter? *See* Final Action, p. 3.

2. Are Claims 1-6, 8-12, 14, 15, and 17-20 properly rejected under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent Application Publication No. 1002/0138582 to Chandra et al. (hereinafter "Chandra")? *See* Final Action, p. 3.

Argument

I. Introduction

Claims 8-12, 14 and 15 stand rejected as allegedly being directed to nonstatutory subject matter. 35 U.S.C. § 101 provides:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

MPEP § 2106 IV.B states:

> To properly determine whether a claimed invention complies with the statutory invention requirements of 35 U.S.C. 101, USPTO personnel must first identify whether the claim falls within at least one of the four enumerated categories of patentable subject matter.
>
> In many instances it is clear within which of the enumerated categories a claimed invention falls. The scope of 35 U.S.C. 101 is the same regardless of the form or category of invention in which a particular claim is drafted. *AT&T*, 172 F.3d at 1357, 50 USPQ2d at 1451. . .
>
> . . . The burden is on the USPTO to set forth a *prima facie* case of unpatentability. Therefore if USPTO personnel determine that it is more likely than not that the claimed subject matter falls outside all of the statutory categories, they must provide an explanation . . .

Claims 1 and 3-18 stand rejected as allegedly anticipated. To anticipate a claim, the reference must teach every element of the claim. M.P.E.P. § 2131. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

II. Claims 8-12, 14 and 15 relate to statutory subject matter

Claims 8-12, 14 and 15 stand rejected as allegedly directed to nonstatutory subject matter. Final Action, p. 3. In particular, the Final Action states "[t]he language of the claims raises the question at to whether or not the claims may be implemented in software alone." Office Action, p. 3. Applicants submit that this reasoning fails to provide an adequate basis for a *prima facie* showing of non-statutory subject matter.

Claim 8 recites:

> Computer apparatus configured to provide:
> a user interface builder object operative to identify a user interface template that implements a task pattern for a model class, to identify a model adapter that provides an interface for an application that conforms to the

> model class, and to create a user interface for the application from the identified user interface template and the identified model adapter.

Thus, Claim 8 clearly recites a "***computer apparatus***," *i.e.*, there is no question as to "whether the claims may be implemented in software alone." Accordingly, Appellants respectfully submit that the §101 of Claim 8 is erroneous and should be reversed. Appellants further submit that the §101 rejections of Claims 9-12, 14 and 15 should be reversed for at least similar reasons.

III. Claims 1-6, 8-12, 14, 15, and 17-20 are patentable over Chandra

A. Independent Claims 1, 8, 14 and 17 are patentable

Independent Claims 1, 8, 14 and 17 stand rejected as allegedly anticipated by Chandra. Final Action, p. 3. Claim 1 recites:

> A method of providing a user interface for an application, the method comprising:
> identifying a user interface template that implements a task pattern for a model class;
> identifying a model adapter that provides an interface for the application that conforms to the model class; and
> creating a user interface for the application from the identified user interface template and the identified model adapter.

For example, as explained at pages 8 of the specification, a task pattern commonly used with enterprise information systems (EIS) and relational databases is a "search-and-browse" pattern. According to some embodiments of the present invention, a user interface template may be developed for a model class that supports this task pattern. A model adapter for a particular application, such as a particular relational database, may be combined with the user template to create a user interface for the application that implements the particular task pattern.

The Final Action cites paragraphs 394 and 399-400 as teaching the recitations of Claim 1. Final Action, p. 3. Chandra describes transportable applications that may be used for communication and collaboration. Chandra, paragraph 34. The transportable applications may be distributed using transport mechanisms similar to e-mail. Chandra, paragraph 36. The transportable applications may be structured as

"building blocks," which are "functional units of a transportable application" and which "encapsulates business logic, data and presentation." Chandra, paragraph 329.

The cited paragraph 394 of Chandra describes Presenter objects ("presenters") that work with Presenter Data Objects (PDOs) that "contains all data of a building block that can possible can be displayed on any device in any form, and all associated interaction pathways." Chandra, paragraph 391. A presenter uses a PDO to do a presentation, for example, an HTML presenter presents data in a PDO in an HTML format. Chandra, paragraph 391. Presenters may be "generic," *i.e.*, may work with PDOs that conform to a particular interface, or a converter may be used to map a specific PDO to such a generic presenter object. Chandra, paragraph 394.

The cited paragraph 399 of Chandra describes that, when a building block is to be presented, an application server requests the building block to construct a specific PDO. The cited paragraph 400 of Chandra indicates that the application server then instantiates the appropriate custom presenter, generic presenter or generic presenter/converter combination, and passes the instantiated presenter to a Java Server Page that uses the presenter to render data from the PDO.

As noted in Appellants' Amendment filed January 8, 2007, there appears to be nothing in this description that corresponds to "identifying a user interface template that implements a task pattern for a model class," as the cited material from Chandra includes no discussion of anything corresponding to "task patterns" for model classes. Moreover, the cited material also appears to lack anything corresponding to the cited "model adaptor," as the generic presenter adaptor described in the cited material is for adapting a generic presenter to a particular data object, not an adaptor for adapting an application, such as a relational database or EIS system, to conform to a model class.

In response to these arguments, the Final Action merely states "the presenters and the converter correspond to the task pattern and the model adapter." Final Action, p. 2. This does not rebut Appellants' arguments. Unlike Chandra's "presenter," the recited "task pattern" is not an object or other functional component. Rather, Claim 1 recites "a user interface template that implements a task pattern for a model class," *i.e.*, the user interface template implements a task pattern associated with the model class, for example, the "search and browse" pattern described above. Chandra's presenters are "building block presenter objects" (see Chandra, paragraph 391) and,

therefore, are not "task patterns" as recited in Claim 1. Appellants further note that Chandra's presenters are also not "user interface templates," because they do not appear to be generic ***templates*** from which specific user interfaces are generated. Rather, these presenters are specific objects.

Appellants further submit that the "converter" described in Chandra also does not correspond to the recited "model adapter." Chandra's "converter" maps a specific presentation data object (PDO) - an object that contains data to be presented - to a generic presenter object that presents data contained in the PDO. In contrast, the recited model adapter "provides an interface for the ***application*** that conforms to the model class," *i.e.*, the model adaptor enables the application to conform to the model class such that it may co-operate with the user interface created from the template to support execution of the task pattern. Chandra's converter does not appear to provide such functionality.

In light of the foregoing, Appellants submit that Chandra does not teach or suggest several of the recitations of independent Claim 1 and, for at least these reasons, Appellants submit that Claim 1 is patentable and that the rejection thereof should be reversed. Appellants further submit that independent Claims 8, 14 and 17 are patentable for at least similar reasons and that the rejections thereof should also be reversed. Applicants submit that dependent Claims 2-6, 9-12, 15 and 18-20 are patentable at least by virtue of the patentability of the respective ones of independent Claims 1, 8, 14 and 17 from which they depend, and that the rejections thereof should be reversed.

B. Claims 3, 10 and 19 are separately patentable

Claim 3 recites "wherein the user interface template defines an abstract portlet, and wherein creating a user interface comprises creating a portlet instance that communicates with the application via the model adapter." In rejecting Claim 3, the Final Action cites paragraphs 368-370, 399 and 400. Final Action, p. 2. As noted in Appellants' Amendment filed January 8, 2007, the cited paragraphs describe use of a "personal portal" that provides a user with access to transportable applications, *e.g.*, membership services, lists of transportable applications received by the user, etc., such that the "portal provides an organized entry point through which a user may

create, organize and send transportable applications and access tools and services for doing so." Chandra, paragraph 367. This appears to have nothing to do with a template that defines an abstract portlet that is instantiated to provide communication "with the application via the model adapter."

In response to these arguments, the Final Action states "Chandra teaches presenters displaying portals and portlets that can use converters [¶368-370, 399-400, 405, 406]." Final Action, p. 2. This does not rebut Appellants' arguments. As discussed above, the "portal" described in Chandra is a "personal portal," *e.g.*, a web page, through which a user accesses the transportable application system described in Chandra. There is no discussion of "portlets", *i.e.*, "visible active components users of [a] portal see within the portal page" (see Specification, p. 6, lines 9 and 10), in Chandra, much less any description of a user interface template that "defines an abstract portlet" or creation of "a portlet instance that communicates with the application via the model adapter." Accordingly, Applicants submit that Chandra does not disclose or suggest the recitations of Claim 3 and, for at least these reasons, Appellants submit that Claim 3 is separately patentable. Appellants submit that at least similar reasons support the separate patentability of Claims 10 and 19. For at least these additional reasons, the rejections of these claims should be reversed.

C. Claims 4, 11 and 20 are separately patentable

Claim 4 recites "wherein creating a portlet instance comprises configuring the portlet instance using an application portlet builder portlet." The Final Action cites paragraphs 428-432 and 441-445 of Chandra as teaching such recitations. Final Action, p. 4. As noted in Appellants' Amendment filed January 8, 2007, these paragraphs refer to an "application builder," but this is a tool for creating transportable applications (see Chandra, paragraph 427), not portlets.

In response to these arguments, the Final Action states "Chandra teaches presenters displaying portals and portlets that can use converters [¶368-370, 399-400, 405, 406]." Final Action, p. 2. This does not rebut Appellants' arguments, as it fails to address the contrasts between the "application builder" described in Chandra and the portlet builder recited in the claims. As discussed above, Chandra is silent about "portlets" and, thus, does not disclose or suggest a "portlet builder." Accordingly,

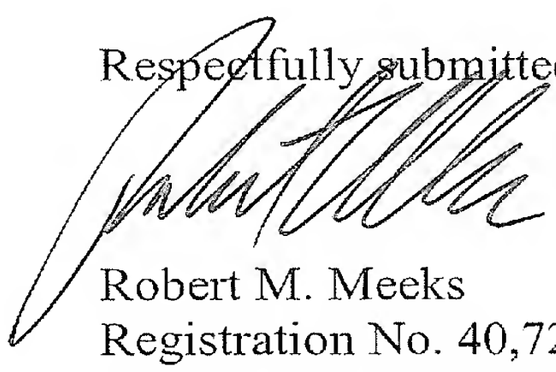
Applicants submit that Chandra does not disclose or suggest the recitations of Claim 4 and, for at least these reasons, Claim 4 is separately patentable. At least similar reasons support the separate patentability of Claims 11 and 20. The rejections of these claims should be reversed for at least these additional reasons.

IV. Conclusion

In light of the above discussion, Appellants submit that the pending claims are directed to patentable subject matter and, therefore, request reversal of the rejections of those claims and passing of the application to issue.

It is not believed that an extension of time and/or additional fee(s) are required, beyond those that may otherwise be provided for in documents accompanying this paper. In the event, however, that an extension of time is necessary to allow consideration of this paper, such an extension is hereby petitioned for under 37 C.F.R. §1.136(a). Any additional fees believed to be due in connection with this paper may be charged to Deposit Account No. 09-0461.

Respectfully submitted,



Robert M. Meeks
Registration No. 40,723
Attorney for Appellants

Customer No. 46589
Myers Bigel Sibley & Sajovec, P.A.
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401

APPENDIX A
Pending Claims USSN Serial No. 10/673,752
Filed September 29, 2003

1. (Original) A method of providing a user interface for an application, the method comprising:

identifying a user interface template that implements a task pattern for a model class;

identifying a model adapter that provides an interface for the application that conforms to the model class; and

creating a user interface for the application from the identified user interface template and the identified model adapter.

2. (Original) A method according to Claim 1, wherein the user interface template comprises a generic view and a generic controller.

3. (Original) A method according to Claim 1, wherein the user interface template defines an abstract portlet, and wherein creating a user interface comprises creating a portlet instance that communicates with the application via the model adapter.

4. (Original) A method according to Claim 3, wherein creating a portlet instance comprises configuring the portlet instance using an application portlet builder portlet.

5. (Original) A method according to Claim 3, wherein identifying a user interface template, identifying a model adapter and creating a portlet instance are performed using an application portlet builder portlet.

6. (Original) A method according to Claim 1, wherein the model adapter provides a business object interface.

7. (Canceled)

8. (Previously Presented) Computer apparatus configured to provide:

a user interface builder object operative to identify a user interface template that implements a task pattern for a model class, to identify a model adapter that provides an interface for an application that conforms to the model class, and to create a user interface for the application from the identified user interface template and the identified model adapter.

9. (Previously Presented) Computer apparatus according to Claim 8, wherein the user interface template comprises a generic view and a generic controller.

10. (Previously Presented) Computer apparatus according to Claim 8, wherein the user interface template defines an abstract portlet, and wherein the user interface builder is operative to create a portlet instance that communicates with the application via the model adapter.

11. (Previously Presented) Computer apparatus according to Claim 10, wherein the user interface builder comprises an application portlet builder portlet operative to configure the portlet instance.

12. (Previously Presented) Computer apparatus according to Claim 8, wherein the model adapter provides a business object interface.

13. (Canceled)

14. (Previously Presented) Computer apparatus configured to provide:

an application;

a model adapter that provides an interface for the application that conforms to a model class; and

a portlet that implements a task pattern for the model class and that communicates with the application via the model adapter according to the interface.

15. (Previously Presented) Computer apparatus according to Claim 14, wherein the model adapter provides a business object interface.

16. (Canceled)

17. (Original) A computer program product for providing a user interface to an application, the computer program product comprising computer readable code embodied in a computer readable storage medium, the computer readable code comprising:

program code configured to provide a user interface builder object operative to identify a user interface template that implements a task pattern for a model class, to identify a model adapter that provides an interface for the application that conforms to the model class, and to create a user interface for the application from the identified user interface template and the identified model adapter.

18. (Original) A computer program product according to Claim 17, wherein the user interface template comprises a generic view and a generic controller.

19. (Original) A computer program product according to Claim 17, wherein the user interface template defines an abstract web portlet, and wherein the code configured to provide a user interface builder comprises code configured to create a concrete web portlet instance that communicates with the application via the model adapter.

20. (Original) A computer program product according to Claim 19, wherein the code configured to create a concrete web portlet instance comprises code configured to provide an application portlet builder portlet operative to configure the portlet instance.

APPENDIX B – EVIDENCE APPENDIX

(NONE)

APPENDIX C – RELATED PROCEEDINGS

(NONE)